

**XEROX**  
**BUSINESS SYSTEMS**  
*System Development Division*  
October 27, 1977

To: Distribution

From: Dick Sweet

Subject: Mesa Language Working Group minutes

The Language Working Group met on 25 October, 1977. The first item considered was an ordering of priorities for language changes for Mesa 4.

*First Priority Group*

- Open Procedures
- Monitors
- Long and Based Pointers
- String Constants

*Second Priority Group*

- Sequences
- Allocation
- Mutable variant records
- Pointer Arithmetic
- Default Parameters and Fields
- Ports
- Field Descriptors
- Included Identifiers
- Long Integers, Real

*Third Priority Group*

- Machine Dependent Records
- Main body as a Procedure
- Painted Integers
- Declaration Syntax (anywhere in body, after BEGIN, etc.)

*Fourth Priority Group*

- LOOP statement
- Coerce clause

It was decided that someone (Ed, John, and Dick?) should write a proposal for sequences, allocation, mutable, pointer arithmetic, and string constants, which are thought to be interconnected. The previous proposals for long pointers and based pointers should be dusted off and recirculated.

A discussion of Open Procedures followed, with the following points made:

- Some reorganization of the Symbol Table will obviously have to be made.

- Where should a procedure be specified as open? At the declaration, it is declared as potentially inline, along with the default way of calling. A call can specify a means of calling other than the default.

**XEROX SDD ARCHIVES**  
I have read and understood  
Pages \_\_\_\_\_ To \_\_\_\_\_  
Reviewer \_\_\_\_\_ Date \_\_\_\_\_  
# of Pages \_\_\_\_\_ Ref. 115DD-372

What does it mean to EXPORT an open coded procedure? One would have to have a body for the procedure and EXPORT a closed instance. As I recall, the automatic generation of a body was not a popular idea.

What about an open coded procedure whose body is in a DEFINITIONS module when someone wants to call it in a closed manner? There should be some way for an implementer module to say "Put the code here."

It was noted that the debugger's fine grain table is not well suited to setting breaks inside an open coded instance.

Likewise, if there are copies of parameters and/or local variables of the open coded procedure, the debugger may have trouble. This also goes for the general expansion of "contexts" provided for by new declaration syntax on a compound statement basis.

Several consecutive open coded calls should share storage for local variables.

When should the compiler make local copies of input parameters? This requires global flow analysis or more to do in the general case. It was deemed safe not to copy in the following case.

1. The variable is not assigned to or subject to the @ operator.
2. There are no field assignments using pointers.
3. There are no procedure calls.

There was a discussion of other fuzzy features of local variables that would make them substitutable, but no real conclusions. It would be nice to allow a terminal procedure call, so that the Pilot people could repackage parameter lists efficiently, e.g.

```
p: PROCEDURE [h: Handle, a,b,c,...] = INLINE
  BEGIN
    h.p[a,b,c,...];
  END;
```

It has also been observed that the ability to SHARE program modules makes all global variables suspect after an external procedure call.

### *Syntax Discussion*

It was agreed that the attribute of being open coded should appear on the body. The popular choice of keyword was INLINE. Syntax was needed for the following cases

1. At the declaration, to specify INLINE, and to specify the default method of calling (inline or out-of-line). Butler suggested INLINE ON DEMAND for a default of out-of-line.
2. At the declaration, to specify that a body is also to be compiled. This would presumably be automatic if the default calling method is out-of-line.
3. In an implementer module, when the body is in a definitions module, something to specify "Put a body here." Presumably the same keyword would work for 2 and 3.
4. At the call site, whether to call in or out of line. No particular resolution.